Conditional Euler Generator

Carl REMLINGER[1,2,3], Joseph MIKAEL[2,3], Romuald ELIE[1]
[1]Université Gustave-Eiffel, [2]EDF Lab, [3]FiME Lab

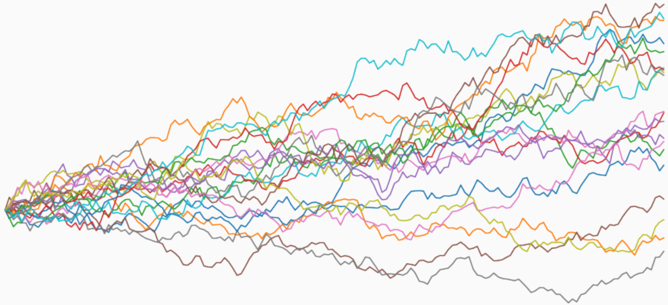
Journées ateliers du Laboratoire FiME
22-23 septembre 2021

Introduction

Time series generation

- What is it?
- Why is it interesting for the industry?
- Why now?

Time what?



Generate new data as faithful as possible to the real data.

The model has to capture marginal distributions *and* temporal dynamics.

Why?

Time series modelling is needed by most of decision making processes:

- **Risk management:** hedging, stress testing, investment decision
- **Asset management:** power plant optimization, portfolio management

Why now?

Deep Reinforcement Learning allows to lift some of the hypothesis required by stochastic control tools.

We are now able to deal with problems in dimension 100 but we do not have satisfactory enough models to use this breakthrough.

$\Rightarrow$ Better models for risk factors are required.

Global approach

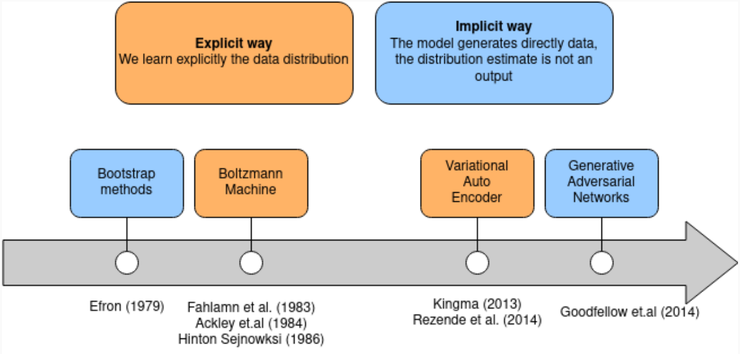
Approach

Model-based

VS

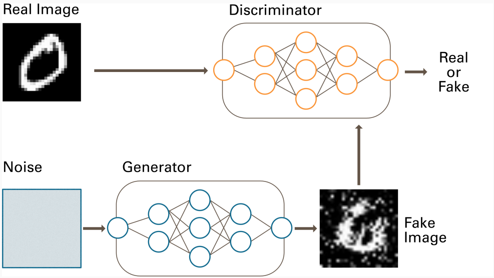
Model-free

How to do it?



Explicit way
We learn explicitly the data distribution
Implicit way
The model generates directly data, the distribution estimate is not an output
Bootstrap methods
Boltzmann Machine
Variational Auto Encoder
Generative Adversarial Networks
Efron (1979)
Fahlamn et al. (1983)
Ackley et.al (1984)
Hinton Sejnowksi (1986)
Kingma (2013)
Rezende et al. (2014)
Goodfellow et.al (2014)

Generative Adversarial Networks [GPAM+14]



$$\inf_{\theta} \sup_{\varphi} \ell\left(d_\varphi(x), d_\varphi(g_\theta(z))\right)$$

Question: Which loss ℓ to be minimized for faithful generations?

Question: How to transform the data for faithful generations?

About loss

The Wasserstein distance (or Kantorovich-Rubinstein) between probability distributions

$$\mathcal{W}_{c,p}(\mu,\nu) \triangleq \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{E \times F} c(x,y)^p d\pi(x,y) \right)^{\frac{1}{p}}$$

Examples on images with Wasserstein GAN [ACB17] or Sinkhorn GAN [GPC18].

$$\mathcal{W}_c(\mu,\nu) \triangleq \inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}^{\pi}\left[c(x,y)\right]$$

Recent applications of GAN for time series

- COT-GAN [XWMA20]: uses a time Adapted Wasserstein distance for continous time
- Signatures [CK16]: embeds path trajectories with signatures
- DVD GAN [CDS19]: combines a spacial and a temporal discriminator
- Time Series GAN [YJvdS19]: applies deep embedding on a latent space while training a GAN

COT-GAN (1/2): Adapted Wasserstein distance [BVBBE20]

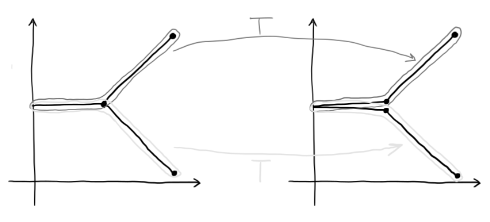


Figure 1: Close in Wasserstein, very different for utility maximization.

We want the transport T (s.a. $T(\mu) = \nu$) to be non-anticipative, i.e. $(T_1(X_1, X_2), T_2(X_1, X_2)) \sim (Y_1, Y_2)$.

Causal Wasserstein distance:

$$\mathcal{W}_c^{\text{causal}}(\mu, \nu) \triangleq \inf_{\pi \in \Pi^{\text{causal}}(\mu,\nu)} \mathbb{E}^{\pi}\left[c(x, y)\right]$$

COT-GAN (2/2): Application of Causal OT [XWMA20]

Definition: Adapted Wasserstein distance in continous time setup

$$\mathcal{AW}_p(\mu, \nu) \triangleq \inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_\pi \left[(x-y)_t^{p/2} + MV_t(|x-y|^p) \right]^{1/p}$$

where MV denotes the mean variation, i.e. $MV_T(Z) = \sup_\Delta \sum_{t_i \in \Delta} \left| \mathbb{E}[Z_{t_{i+1}} - Z_{t_i} | \mathcal{F}_{t_{i+1}}] \right|$

Adapted cost function:

$$c_\varphi^{\mathcal{K}}(x, y) = c(x, y) + \sum_j \sum_t h_{\varphi_1,t}^j(y) \Delta_{t+1} M_{\varphi_2,t}^j(x)$$

where h, M are neural networks parametrized by $\varphi = (\varphi_1, \varphi_2)$.

Causal OT GAN:

$$\inf_\theta \sup_\varphi \mathcal{W}_{c_\varphi}(\mu, \nu_\theta)$$

Signatures [CK16]

Definition: The signature of a path $X : [a, b] \to \mathbb{R}^d$, denoted $S(X)_{a,b}$ is a collection of all the iterated integrals of X:

$$S(X)_{a,b} = (1, S(X)^{(1)}, ..., S(X)^{(d)}, S(X)^{(1,1)}, S(X)^{(1,2)}, ...).$$

where the *k-fold iterated integral* of X is:

$$S(X)_{a,b}^{(i_1,...i_k)} = \int_{a<t_k<b} ... \int_{a<t_1<t_2} dX_{t_1}^{i_1} ... dX_{t_k}^{i_k}$$

	d=2	d=3	d=6
k=1	2	3	6
k=3	6	12	42
k=5	62	363	9330
k=7	254	3279	335922

Table 1: Size of $S^{(k)}(X)$ for different coefficient k and dimension d of X

Some applications SigCWGAN [NSW+20], SigCVAE [BHL+20]

DVD GAN [CDS19]

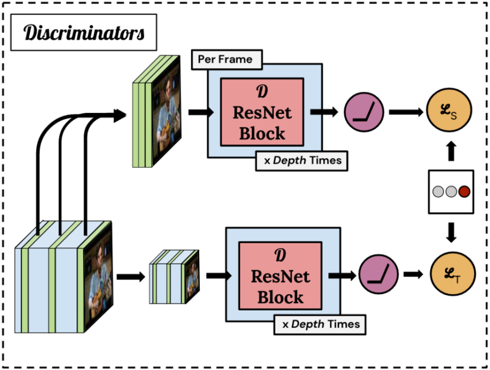


Figure 2: Time Series GAN scheme

While the *spacial* discriminator focuses on time marginals and critics images in high resolution, the *temporal* one considers the full sequence of images in low resolution.

Time Series GAN [YJvdS19]

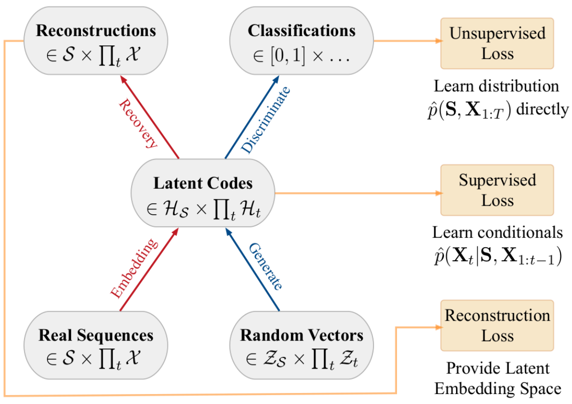


Figure 3: Time Series GAN scheme

Problems

- No control on generations
- Learning marginals is not enough
- Learning joint distribution too (for now)
- Few theoretical results

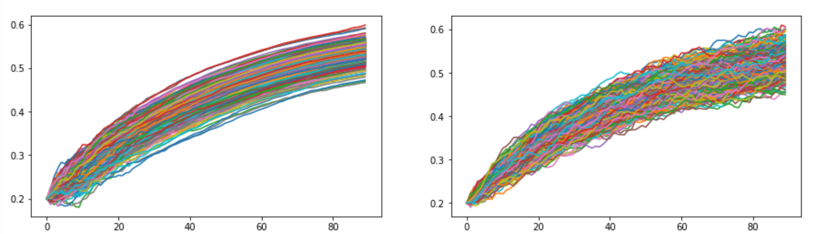


Figure 4: Left: Unsatisfactory generations from GAN **Right:** Reference (Ornstein-Uhlenbeck).

Conditional Euler Generator

Main Contributions

- New time series generator, called CEGEN, combines an *Euler structure* with a dedicated *loss on conditional distributions*
- A theoretical result ensuring a bound for the process parameter estimation error.
- Numerical study on synthetic and various real world data sets: accurate correlation structure up to dimension 20
- Model-based mixture with transfer learning when few data is available.

CEGEN - Model

We are given i.i.d. samples of a time series, viewed as a Itô process $X = (X_t)_t$:

$$dX_t \quad = \quad b_X(t, X_t)dt + \sigma_X(t, X_t)dW_t$$

Deep Euler representation

$$\begin{aligned} X_{t_i+\Delta t} &= X_{t_i} + b_X(t_i, X_{t_i})\Delta t + \sigma_X(t_i, X_{t_i})\Delta W_{t_i} \qquad (1) \\ Y^\theta_{t_i+\Delta t} &= Y^\theta_{t_i} + g^b_\theta(t_i, Y^\theta_{t_i})\Delta t + g^\sigma_\theta(t_i, Y^\theta_{t_i})Z_{t_i} \qquad (2) \end{aligned}$$

where $i = 1..N$ and $Z_{t_i} \sim \mathcal{N}(0, \Delta t I_d)$.
Our goal is to learn g^b_θ and g^σ_θ, approximated by neural networks, so that the distributions of the processes Y^θ and X are close.

CEGEN - Model

We are given i.i.d. samples of a time series, viewed as a Itô process $X = (X_{t_i})_{i=1...N}$:

$$dX_t = b_X(t, X_t)dt + \sigma_X(t, X_t)dW_t$$

Deep Euler representation

$$X_{t_i+\Delta t} = X_{t_i} + b_X(t_i, X_{t_i})\Delta t + \sigma_X(t_i, X_{t_i})\Delta W_{t_i} \quad (3)$$

$$Y^\theta_{t_i+\Delta t} = Y^\theta_{t_i} + g^b_\theta(t_i, Y^\theta_{t_i})\Delta t + g^\sigma_\theta(t_i, Y^\theta_{t_i})Z_{t_i} \quad (4)$$

where $Z_{t_i} \sim \mathcal{N}(0, \Delta t I_d)$.
Our goal is to learn g^b_θ and g^σ_θ, approximated by neural networks, so that the distributions of the processes Y^θ and X are close.

CEGEN - Model

We are given i.i.d. samples of a time series, viewed as a Itô process $X = (X_{t_i})_{i=1...N}$:

$$dX_t = b_X(t, X_t)dt + \sigma_X(t, X_t)dW_t$$

Deep Euler representation

$$\begin{aligned} X_{t_i+\Delta t} &= X_{t_i} + b_X(t_i, X_{t_i})\Delta t + \sigma_X(t_i, X_{t_i})\Delta W_{t_i} && (5) \\ Y^\theta_{t_i+\Delta t} &= Y^\theta_{t_i} + g^b_\theta(t_i, Y^\theta_{t_i})\Delta t + g^\sigma_\theta(t_i, Y^\theta_{t_i})Z_{t_i} && (6) \end{aligned}$$

where $Z_{t_i} \sim \mathcal{N}(0, \Delta t I_d)$.
Our goal is to learn g^b_θ and g^σ_θ, approximated by neural networks, so that the distributions of the processes Y^θ and X are close.

CEGEN - Loss

We build a loss function that compares the conditional distribution $\mathcal{L}(Y^\theta_{t_{i+1}} \,|\, Y^\theta_{t_i})$ with $\mathcal{L}(X_{t_{i+1}} \,|\, X_{t_i})$ for each time step t_i.

Wasserstein-2 distance definition in the Gaussian case:

$$\mathcal{W}_2^2(\mathcal{L}(X), \mathcal{L}(Y)) \triangleq \|\mathbb{E}[X] - \mathbb{E}[Y]\|_2^2 + \mathcal{B}^2(Var(X), Var(Y))$$

where $\mathcal{B}$ is the Bures metric:

$$\mathcal{B}^2(A, B) \triangleq Tr(A) + Tr(B) - 2Tr(A^{\frac{1}{2}} B A^{\frac{1}{2}})^{1/2}$$

Motivation

Let $t_i \in \{t_0, \ldots, t_n = T\}$. For $z \in \mathbb{R}^d$.

If $\forall \varepsilon > 0$,

$$\mathcal{W}_2^2 \left(\mathcal{L}(X_{t_{i+1}} | X_{t_i} = z), \mathcal{L}(Y^\theta_{t_{i+1}} | Y^\theta_{t_i} = z) \right) \leq \varepsilon$$

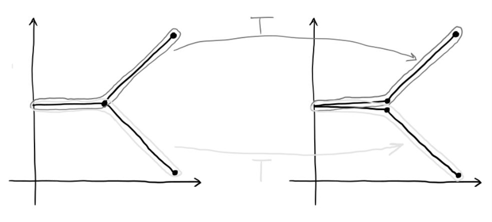
Then,

$$\| b_X(t_i, z) - b_Y(t_i, z) \|_2^2 \quad \leq \quad \frac{\varepsilon}{(\Delta t)^2}$$

$$\begin{aligned} \| \sigma_X(t_i, z) - \sigma_Y(t_i, z) \|_2^2 &\leq \frac{\varepsilon}{\Delta t} \quad \text{if} \quad d = 1 \\ &\leq \frac{2\alpha^2 \varepsilon}{\Delta t} \quad \text{if} \quad d > 1 \text{ and} \\ &\qquad\qquad Tr\left(\sigma_X^2(t_i, z)\right) = Tr\left(\sigma_Y^2(t_i, z)\right) = \alpha \end{aligned}$$

In pratice...

In pratice it is tricky to conditionate at the very same point.



$$\hat{\mathcal{W}}_2\left(\mathcal{L}(X_{t_2}|X_{t_1}\in I),\mathcal{L}(Y^{\theta}_{t_2}|Y^{\theta}_{t_1}\in I)\right)$$

$$\approx$$

$$\mathcal{W}_2\left(\mathcal{L}(X_{t_2}|X_{t_1}=z),\mathcal{L}(Y^{\theta}_{t_2}|Y^{\theta}_{t_1}=z)\right)$$

for ball I of radius Δx "small".

Proposition

Assume that $\sigma^2_X(t_i,.)$, $\hat{\sigma}^2_{Y^\theta}(t_i,.)$ *are strictly positive and, together with* $b_X(t_i,.)$ *and* $\hat{b}_{Y^\theta}(t_i,.)$, *K-Lipschitz in their second coordinate. For* $t_i \in \mathcal{T}$, *let* $(I_k)_k$ *be a regular partition covering* $Supp(X_{t_i}) \cup Supp(Y_{t_i})$ *with mesh size* Δx. *Let* $\varepsilon > 0$.
If for any k,

$$\hat{\mathcal{W}}_2^2 \left(\mathcal{L}(X_{t_{i+1}} | X_{t_i} \in I_k), \mathcal{L}(Y^\theta_{t_{i+1}} | Y^\theta_{t_i} \in I_k) \right) \leq \varepsilon$$

then, for z *in the partition* I_k

$$\|b_X(t_i,z) - b_Y(t_i,z)\|_2 \quad \leq \quad \frac{\sqrt{\varepsilon} + \Delta x}{\Delta t} + 2K\Delta x$$

Furthermore

$$\|\sigma_X(t_i,z) - \sigma_Y(t_i,z)\|_2 \leq \sqrt{\frac{\varepsilon}{\Delta t}} + 2K\Delta x \quad \text{if} \quad d = 1$$

$$\|\sigma_X(t_i,z) - \sigma_Y(t_i,z)\|_2 \leq \sqrt{\frac{2\alpha^2\varepsilon}{\Delta t}} + 2K\Delta x \quad \text{if} \quad d > 1 \text{ and } Tr\left(\sigma^2_X(t_i,z)\right) = Tr\left(\sigma^2_Y(t_i,z)\right) = \alpha$$

Compute Bures metric

$$\mathcal{B}^2(A, B) \triangleq Tr(A) + Tr(B) - 2\,Tr(A^{\frac{1}{2}} B A^{\frac{1}{2}})^{1/2}$$

Need to compute matrix roots and their inverses: we use batch approximation.

Algorithm 1 Newton-Schulz Square Root

Input: PSD matrix $\mathbf{A}$, $\epsilon > 0$

- $\mathbf{Y} \leftarrow \frac{\mathbf{A}}{(1+\epsilon)\|\mathbf{A}\|}$, $\mathbf{Z} \leftarrow \mathbf{I}$
- **while** not converged **do**
  - $\mathbf{T} \leftarrow (3\mathbf{I} - \mathbf{Z}\mathbf{Y})/2$
  - $\mathbf{Y} \leftarrow \mathbf{Y}\mathbf{T}$
  - $\mathbf{Z} \leftarrow \mathbf{T}\mathbf{Z}$
- **end while**
- $\mathbf{Y} \leftarrow \sqrt{(1+\epsilon)\|\mathbf{A}\|}\,\mathbf{Y}$
- $\mathbf{Z} \leftarrow \frac{\mathbf{Z}}{\sqrt{(1+\epsilon)\|\mathbf{A}\|}}$

Output: square root $\mathbf{Y}$, inverse square root $\mathbf{Z}$

GPU friendly

Root + Inverse: two for one!

Figure 5: Algorithm to compute matrix roots and their inverses [MC18]

Some results

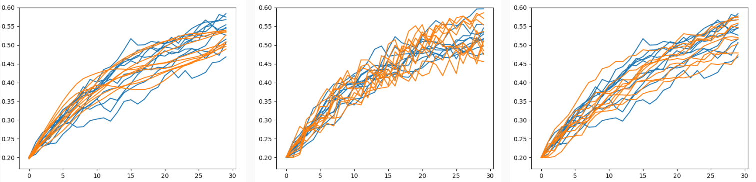


Figure 6: Ornstein-Uhlenbeck process samples (blue) and generations (orange) from COTGAN, TSGAN and CEGEN.

Real world time series

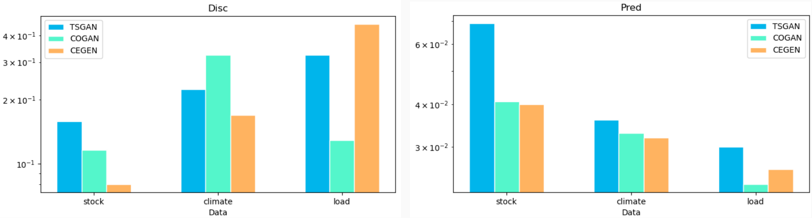
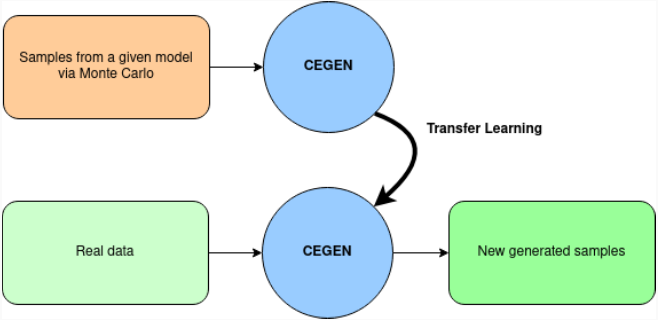


Figure 7: Left: Discriminative Score **Right:** Predictive Score

Transfer learning

We propose to start the training with a reasonably wrong (known) model and to finish up the training with the few real data samples.



Transfer learning tests show how model-free method can rely on a proven simulation model without replacing it completely.

Conclusion

- Generation of realistic time series is particularly interesting for energy companies or the financial sector
- Generators should not be left free, as this could lead to damageable consequences
- Contrarily to the existing model-free approaches, we impose an Euler structure and consider a conditional loss
- Theoretical bound for the estimation error of the process parameters
- CEGEN manages to get good behavior for synthetic as well as for real data

Questions?

Bibliography i

Martin Arjovsky, Soumith Chintala, and Léon Bottou.
Wasserstein gan.
arXiv preprint arXiv:1701.07875, 2017.

Hans Buehler, Blanka Horvath, Terry Lyons, Imanol Perez Arribas, and Ben Wood.
Generating financial markets with signatures.
Available at SSRN, 2020.

Julio Backhoff-Veraguas, Daniel Bartl, Mathias Beiglböck, and Manu Eder.
Adapted wasserstein distances and stability in mathematical finance.
Finance and Stochastics, 24:601–632, 2020.

Aidan Clark, Jeff Donahue, and Karen Simonyan.
Adversarial video generation on complex datasets.
arXiv preprint arXiv:1907.06571, 2019.

Ilya Chevyrev and Andrey Kormilitzin.
A primer on the signature method in machine learning.
arXiv preprint arXiv:1603.03788, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Aude Genevay, Gabriel Peyré, and Marco Cuturi.
Learning generative models with sinkhorn divergences.
In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617, 2018.

Boris Muzellec and Marco Cuturi.
Generalizing point embeddings using the wasserstein space of elliptical distributions.
In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10258–10269, 2018.

Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao.
Conditional sig-wasserstein gans for time series generation.
arXiv preprint arXiv:2006.05421, 2020.

Tianlin Xu, Li K Wenliang, Michael Munn, and Beatrice Acciaio.
Cot-gan: Generating sequential data via causal optimal transport.
arXiv preprint arXiv:2006.08571, 2020.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar.
Time-series generative adversarial networks.
2019.